

# Hidden Message

## Step One

Given a string that has been converted to Morse code, find all of the possible and unique sequences of remaining tokens after removing the second message from the first. There are 3 different types of tokens in the Morse code message.

- Dot (\*)
- Dash (-)
- Blank (\_)

Every letter in the message is separated by a single blank character and every word is separated by 3 blank characters.

### Example:

- Given: AB  
○ \* - \_ - \* \* \*
- Remove: R  
○ \* - \*
- This can be done 6 different ways.  
○ X X \_ - X \* \*  
○ X X \_ - \* X \*  
○ X X \_ - \* \* X  
○ X - \_ X X \* \*  
○ X - \_ X \* X \*  
○ X - \_ X \* \* X
- But this results in only 2 possible unique sequences of remaining tokens.  
○ \_ - \* \*  
○ - \_ \* \*
- So the final result for this example would be 2.

Write a program that can calculate all of the deletion paths for removing one Morse code message from another. This program should be able to calculate all of the paths in the example below in under 10 seconds and return the total number of paths found. This is a preliminary step for your program, for testing.

- Given: Hello World  
○ \* \* \* \* \_ \* \_ \* - \* \* \_ \* - \* \* \_ \_ \_ \_ \* \_ \_ \_ \_ \* - \* \_ \* - \* \* \_ \* \* \*
- Remove: Help  
○ \* \* \* \* \_ \* \_ \* - \* \* \_ \* - \_ \*
- ANSWER: 1311

## Step two

Find the possible deletion paths where you must remove a second message from the remaining tokens in the original message after removing the first phrase, including blank tokens. All remaining tokens would be kept in the same order to find a second phrase. Return all of the possible and unique sequences of remaining tokens after removing both phrases.

In the example above where R is removed from AB notice that there are 6 delete paths but only 2 unique sequences of remaining tokens.

### Example:

- Given: ABCD
  - \* \_ \_ \* \* \* \_ \_ \* \_ \_ \* \*
- Remove: ST
  - \* \* \* \_ \_
- Then Remove: ZN
  - \_ \_ \* \* \_ \_ \*
- One solution path would look like:
  - Start:
    - \* \_ \_ \_ \* \* \* \_ \_ \* \_ \_ \* \_ \_ \* \*
  - Remove ST:
    - x \_ \_ \_ x x \* x x \* \_ \_ \* \_ \_ \* \*
  - Then Remove ZN:
    - x x \_ \_ x x x x x x x x \_ \_ \* x x x \*
  - The set of remaining characters
    - \_ \_ \* \*
- There are 5 sequences of remaining characters for this example:
  - \_ \_ \* \*
  - \_ \_ \* \_ \*
  - \_ \_ \* \*
  - \* \_ \_ \*
  - \* \_ \_ \*

Expand your program to find all of the possible sequences of remaining characters after removing 2 hidden Morse code messages from an original message. This program should be able to calculate all of the sequences in the example below in less than 60 seconds and return the total number of distinct and valid sequences found. This will be the final expected output from the program.

- Given: The Star Wars Saga
  - \_ \_ \* \* \* \* \* \_ \_ \* \* \* \_ \_ \* \_ \_ \* \_ \_ \* \_ \_ \* \* \* \_ \_ \* \* \* \_ \_ \* \_ \_ \* \_ \_
- Remove: Yoda
  - \_ \* \_ \_ \_ \_ \_ \* \* \* \_ \_ \*

- And Remove: Leia
  - \*\_\*\* \*\_\*\* \*\_
- Expected Answer: 11474

## Program Specifications

- Input: Three command-line arguments denoting the original message, the first hidden message, and the second hidden message. The three messages will be in Morse code using the representation described here.
- Expected Output: Total number of distinct and valid remaining token sequences in the original message.
- Bounds: Original Message less than 100 Morse code characters.